# Efficient Small and Overlapping Target Detection in Underwater Images

Victor Sineglazov[1,2], Mykhailo Savchenko[1], Volodymyr Lytvynenko[3]

[1] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Prospect Beresteiskyi, Kyiv, 03056, Ukraine
[2] National Aviation University, 1, Prospect Liubomyra Huzara, Kyiv, 03058, Ukraine
[3] Kherson National Technical University, 24, Beryslavske Shose, Kherson, 73008, Ukraine

## Abstract

The paper aims to create a methodology for resource-efficient small and overlapping target detection in underwater images. Hybrid object detection neural network based on YOLOv8 with novel light-weighting technique, enhanced feature map upsampling method and effective non-maximal suppression strategy are proposed to reduce computational complexity and boost localization and classification accuracy of small and overlapping underwater targets. The object detection model based on the methodology from this study is tested on Underwater Target Detection Algorithm Competition 2020 dataset with severe class imbalance and high number of small overlapping targets. Our findings show that the proposed model reaches higher accuracy than existing solutions while being efficient enough to be deployed on edge hardware of autonomous underwater vehicle.

## Keywords[1]

Underwater object detection, autonomous underwater vehicles, neural networks, deep learning, soft non-maximum suppression

## 1. Introduction

Autonomous Underwater Vehicles (AUVs) with object detection functionality are actively used across various domains, leveraging their capabilities for diverse underwater tasks. In marine biology, AUVs facilitate the research and monitoring of biodiversity, providing critical data on marine species and their habitats. Underwater archaeology benefits from AUVs in the search for submerged cultural artifacts and shipwrecks, aiding in the preservation of historical heritage. The oil and gas industry utilizes AUVs to explore underwater topography, optimizing route planning for pipelines and drilling operations. Additionally, AUVs play a crucial role in ecology by monitoring pollution levels, in military applications for demining and threat detection, and in emergency services for exploring wreckages and conducting search and rescue operations.

Despite the advancements in deep learning-based object detection, these systems often struggle to perform adequately in underwater environments. Object detection in AUV imagery faces unique challenges such as the presence of small targets that are difficult to discern, targets overlapping with each other, and the tendency for targets to blend into complex backgrounds like rocks and coral reefs. Targets may be partially obscured by mud, rocks, or other underwater structures, and they may appear in different scales due to varying distances and depths. Additionally, targets can have irregular shapes, such as seaweed, or may become distorted in shape, complicating their identification. Moreover, the datasets images taken by AUVs are often not diverse enough to ensure the deep learning object detection model will generalize well with unseen data.



*ProfIT AI 2024: 4th International Workshop of IT-professionals on Artificial Intelligence (ProfIT AI 2024), September 25–27, 2024, Cambridge, MA, USA*

svm@nau.edu.ua (A. 1); mykhailo_savchenko@outlook.com (A. 2), immun56@gmail.com (A.3)
0000-0002-3297-9060 (A. 1); 0009-0004-4072-5409 (A. 2); 0000-0002-1536-5542 (A.3)

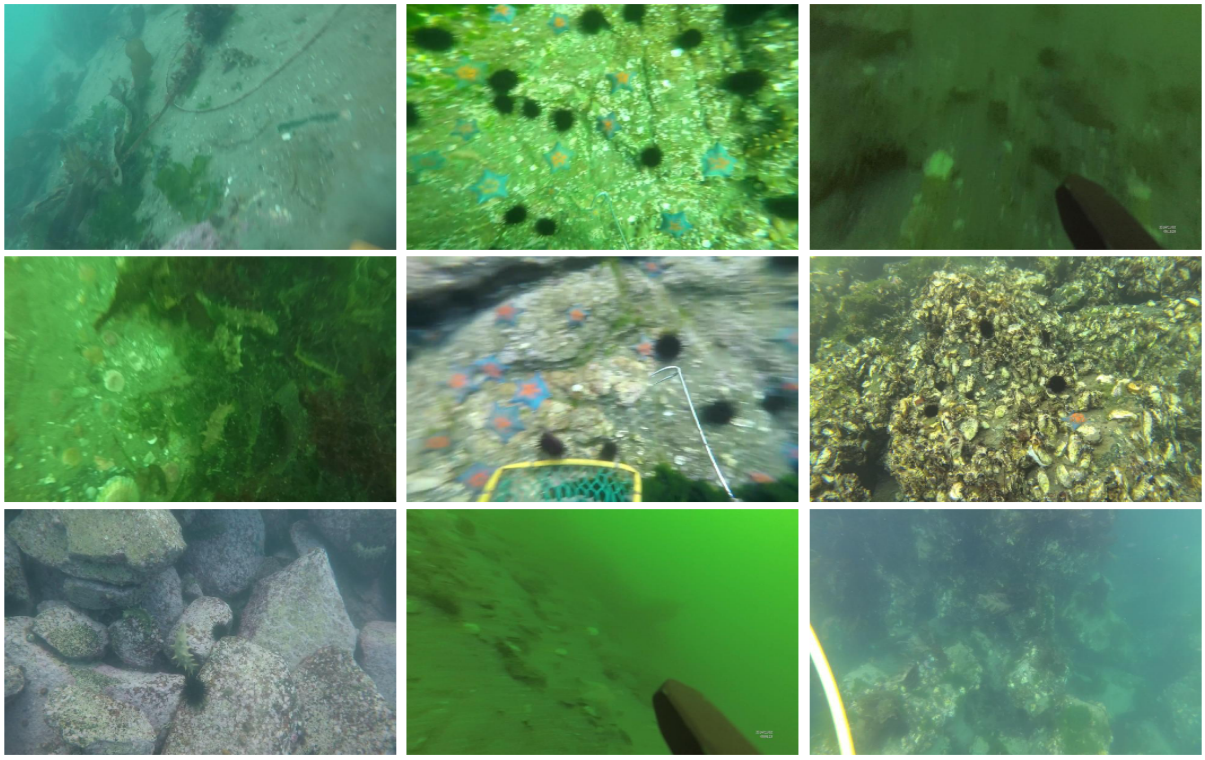

**Figure 1:** Sample images from Underwater Target Detection Algorithm Competition 2020 dataset, demonstrating the presence of tiny, densely-located targets, often subject to cases of occlusion.

Another significant challenge is the computational aspect of object detection in AUV imagery. Processing the data on land centers poses issues related to the communication link between the AUV and the workstation, limiting the operational range and suffering from connection instabilities. Alternatively, running the detection model directly on the AUV demands considerable computational resources, which must be balanced with other essential software for navigation and obstacle avoidance. The need for fast and low-latency computations is often a crucial requirement, particularly in mission-critical scenarios such as military operations.

Given these challenges, there is a pressing need to develop an intelligent system for AUVs that can effectively address visibility issues, detect targets of varying sizes with high accuracy, and operate efficiently under the constraints of underwater environments. Such a system would enhance the capabilities of AUVs, enabling them to perform their tasks more reliably and expanding their utility across various critical applications.

## 2. Related work

In recent years, significant progress has been made in the field of generic object detection and classification, driven by advancements in deep learning and hybrid neural networks [1, 2, 3]. These developments have resulted in highly accurate and efficient models capable of detecting and classifying objects in various environments, including but not limited to medical applications [4], aerial target detection [5] and others. Parallel to this, researchers have been focusing on developing specialized underwater object detection frameworks. These frameworks are specifically tailored to address the unique challenges of underwater scenarios, such as poor visibility, low contrast, and high levels of noise, ensuring reliable and effective detection of objects in marine environments.

### 2.1. Generic object detection models

Anchor-based and anchor-free object detection models are two primary methodologies for deep learning-based object detection. Faster R-CNN [6], SSD [7] and RetinaNet [8] rely on predefined

anchor boxes to localize targets in input images. On the other hand, anchor-free algorithms like YOLOX [9] and FCOS [10] only calculate the center point of the bounding box and position coordinates, making object detection process more straightforward.

Recent advancements of convolutional neural networks have taken these methodologies even further, improving object detection and classification accuracy. Deep Convolutional Neural Networks (CNNs) are used as a backbone of object detection models, extracting object features through multi-layer non-linear transformations. The YOLO (You Only Look Once) series, which relies on these principles, has proven its efficiency and generalization capabilities for diverse object detection tasks.

Multiple enhancements have been introduced throughout the evolution of the YOLO series. YOLOv1 [11] was the first model in this family which offered an effective solution for overcoming limitations of two-stage object detectors by introducing a novel methodology of single shot detection. YOLOv2 [12] proposed batch normalization usage and improved overall network performance by removing dropout. YOLOv3 [13] was the first model of its family which featured Darknet-53 backbone with residual connections and feature pyramid network (FPN) for enhanced aggregation of features. YOLOv4 [14] improved information flow between model layers by introducing improved CSPDarknet53 backbone with cross-partial connections. YOLOv5 [15] was configured to make multi-scale predictions more efficient model structure and automated hyperparameter search for improved model inference and performance. YOLOv7 [16] used ELAN computational block for more effective layer aggregation and offered new model scaling techniques. YOLOv8 [17] uses anchor-free detection mechanism, and enhances feature extraction efficiency with auto augmentation techniques such as cropping and mosaic augmentation.

The architecture of YOLOv8 can be divided into three main components: the backbone, the neck, and the head. The backbone network utilizes convolutional layers, pooling layers, and residual connections to extract detailed features from input images. A key component of YOLOv8 backbone is the C2f block, which enhances gradient flow and feature reuse, making the network deeper and more efficient. The neck, which includes Path Aggregation Network (PANet) and Feature Pyramid Network (FPN) structures, aggregates features from different levels of the backbone, combining high-level semantic information with low-level details to improve multi-scale object detection. This involves upsampling layers and concatenation processes that merge contextual and detailed information. In the head, YOLOv8 predicts bounding boxes, object classes, and confidence scores using predefined anchor boxes, bounding box regression, and class prediction. The detection results are refined through Non-Maximum Suppression (NMS), which removes redundant boxes and retains the highest confidence scores for each detected object.

### 2.2. Underwater object detection frameworks

Underwater object detection frameworks are the models which are specifically tailored for underwater environments and utilize specific methods to gain performance improvements over generic object detection models. These frameworks are used for diverse tasks, including, but not limited to marine biodiversity monitoring [18], identifying and localizing underwater pollution sources and trash [19, 20], human body detection [21].

Single stage detectors, such as YOLO family are most widely used in underwater object detection frameworks for its detection speed and versatility. Numerous improvements have been made to make YOLO faster, computationally cheaper and more accurate for underwater target detection tasks. Zhang et al. [23] introduced a lightweight underwater object detection framework based on YOLOv4 with multi-scale attentional feature fusion. Liu et al. [24] introduced TC-YOLO, combining CLAHE preprocessing, a modified YOLOv5s architecture, and attention mechanisms for improved detection accuracy. Shen et al. [25] proposed the multi-dimensional, multi-functional, and multi-level attention module (mDFLAM) to enhance robustness and generalization in underwater images. Xu et al. [26] introduced SA-FPN, optimizing feature extraction with a scale-aware feature pyramid architecture. Pan et al. [27] developed a modified method based on multi-scale ResNet for improved detection of

objects of various sizes. Wang et al. [28] enhanced YOLOv7 with an image enhancement module and introduced Focal EIOU for bounding box regression loss. Minghua Zhang et al. [29] suggested replacing YOLOv8's original backbone with FasterNet for lower latency, while Guo et al. [30] modified YOLOv8's backbone with FasterNet layers and improved feature pyramid network for better detection capabilities. Sineglazov and Savchenko in [31] suggested object detection model light-weighting methodologies and implemented attention mechanisms into feature fusion process to build a fast and accurate object detector for deploying it directly on AUV hardware.

### 2.3. Limitations of existing methodologies

Despite significant advancements in underwater object detection, several limitations persist in methodologies mentioned above:

- Generic object detection frameworks are optimized for general-purpose tasks and can't handle underwater scenes correctly, yielding too many false negative detections. Underwater object are often small, densely packed and are present in a large count, so there are cases of severe bounding box overlapping. Moreover, the target objects are often located on a complex background (i.e. rocks, mud) or overlapped by larger objects, which makes it more difficult to discern. Therefore, a more domain-aware approach should be considered.
- Specialized underwater object detection frameworks adapt to specific requirements of underwater object detection through the series of preprocessing steps and network topology modifications. However, these modifications are often computationally expensive and are infeasible to deploy on edge platforms like NVIDIA Jetson. These factors directly impact the practicality and efficiency of deploying neural networks in mission-critical underwater applications.

Addressing these limitations requires a more holistic approach to model development. The requirements to a comprehensive underwater object detection framework should include the ability to work with tiny targets with high degree of overlap, the ability to discern object on a complex background or being partially obscured by another object. Additionally, the proposed neural network should be small enough in size and have lower computational requirements, which would make it eligible to be used directly on an integrated hardware of AUV.

## 3. Problem statement

Let $(X_1, Y_1), (X_2, Y_2), \dots, (X_i, Y_i)$ represent the samples from training dataset, $X_i \in \mathbb{R}^{n \times n}$ denote the $i-$th the matrix of RGB image with dimensions $n \times n \times 3$, and let $Y_i$ denote the ground truth annotations for the corresponding image, consisting bounding box coordinates and class labels.

The primary objective is to develop a neural network architecture that is capable of accurately predicting the coordinates of bounding boxes and class probabilities for objects within the input images. This involves training the network to learn optimal weight coefficients that minimize a predefined loss function.

The objective is to build and train a neural network, which will handle the tasks such as bounding box assignment for each object within the given image matrix and prediction of the classes for each object found. In this case, $f_\theta: \mathbb{R}^{n \times n} \to \mathbb{R}^{S \times S \times (B \times 5 + C)}$ represents such a neural network, where parameters are $\theta$, $S$ is the size of a grid, $B$ represents the total number of bounding boxes within the cell of this grid, and $C$ denotes the number of classes in the dataset. The neural network output is the tensor with dimentions $S \times S \times (B \times 5 + C)$, which contains information about predicted bounding boxes and class probabilities for each grid cell.

In such case, the loss function for object detection and classification tasks is defined as:

$$\mathcal{L} = \lambda_{\text{coord}} \cdot \mathcal{L}_{\text{coord}} + \lambda_{\text{conf}} \cdot \mathcal{L}_{\text{conf}} + \lambda_{\text{class}} \cdot \mathcal{L}_{\text{class}}$$

where $\mathcal{L}_{\text{coord}}$, $\mathcal{L}_{\text{conf}}$ and $\mathcal{L}_{\text{class}}$ are the localization, confidence and classification losses respectively, $\lambda_{\text{coord}}$, $\lambda_{\text{conf}}$, $\lambda_{\text{class}}$ are the coefficients to weight the importance of each individual loss in total loss function.

Assuming that box loss is handled by CIoU [32] loss function, multi-label classification loss is handled by binary cross entropy and distribution focal loss [33] is the third term in total loss function, the loss function of described neural network can be rewritten as:

$$\begin{aligned}\mathcal{L} = \frac{\lambda_{\text{box}}}{N_{\text{pos}}} \sum_{x,y} \mathbf{1}_{c^*_{x,y}} \left[ 1 - q_{x,y} + \frac{\left\| b_{x,y} - \hat{b}_{x,y} \right\|_2^2}{\rho^2} + \alpha_{x,y} v_{x,y} \right] \\ + \frac{\lambda_{\text{cls}}}{N_{\text{pos}}} \sum_{x,y} \sum_{c \in \text{classes}} y_c \log(\hat{y}_c) + (1 - y_c) \log(1 - \hat{y}_c) \\ + \frac{\lambda_{\text{dfl}}}{N_{\text{pos}}} \sum_{x,y} \mathbf{1}_{c^*_{x,y}} [-(q_{(x,y)+1} - q_{x,y}) \log(\hat{q}_{x,y})] \\ + (q_{x,y} - q_{(x,y)-1}) \log(\hat{q}_{(x,y)+1})]\end{aligned}$$

where:

$$q_{x,y} = IoU_{x,y} = \frac{\hat{\beta}_{x,y} \cap \beta_{x,y}}{\hat{\beta}_{x,y} \cup \beta_{x,y}}$$

$$v_{x,y} = \frac{4}{\pi^2} \left( \arctan\left(\frac{w_{x,y}}{h_{x,y}}\right) - \arctan\left(\frac{\hat{w}_{x,y}}{\hat{h}_{x,y}}\right) \right)^2$$

$$\alpha_{x,y} = \frac{v}{1 - q_{x,y}}$$

$$\hat{y}_c = \sigma(\cdot)$$

$$\hat{q}_{x,y} = \text{softmax}(\cdot)$$

In this case, $N_{\text{pos}}$ is number of cells featuring an object, $\mathbb{1}_{c^*_{x,y}}$ is an indicator function for the cells with an object, $\beta_{x,y}$ is the position of ground truth bounding box, $b_{x,y}$ is the predicted bounding box for the cell, $\hat{\beta}_{x,y}$ are the center point of the ground truth bounding box coordinates, $y_c$ is the ground truth label for class c for each individual grid cell (x, y) in the input, $q_{(x,y)+1}$ are the nearest left and right predicted boxes IoU which belong to $c^*_{x,y}$, $w_{x,y}$ and $h_{x,y}$ are width and height of the box, and $\rho$ is the diagonal length of the smallest enclosing box covering the predicted and ground truth boxes. Then, each cell determines the best candidate for predicting the bounding box of given object.

For network training, the total loss function $\mathcal{L}$ is optimized with Stochastic Gradient Descent (SGD) with momentum algorithm. SGD with momentum update rule involves these steps:

$$v_{t+1} = \beta v_t + (1 - \beta) \nabla_\theta \mathcal{L}(\theta_t)$$
$$\theta_{t+1} = \theta_t - \eta v_{t+1},$$

where $v_t$ is the velocity term representing the exponentially weighted moving average of past gradients, $\theta_t$ are the parameters of the network at iteration $t$, $\nabla_\theta \mathcal{L}(\theta_t)$ the loss function gradient with respect to the network parameters at iteration $t$, $\eta$ denotes the learning rate, $\beta$ is the momentum term.

## 4. Proposed methodology

To enhance the performance of YOLOv8 for underwater object detection, we introduce several key modifications aimed at improving both efficiency and accuracy.

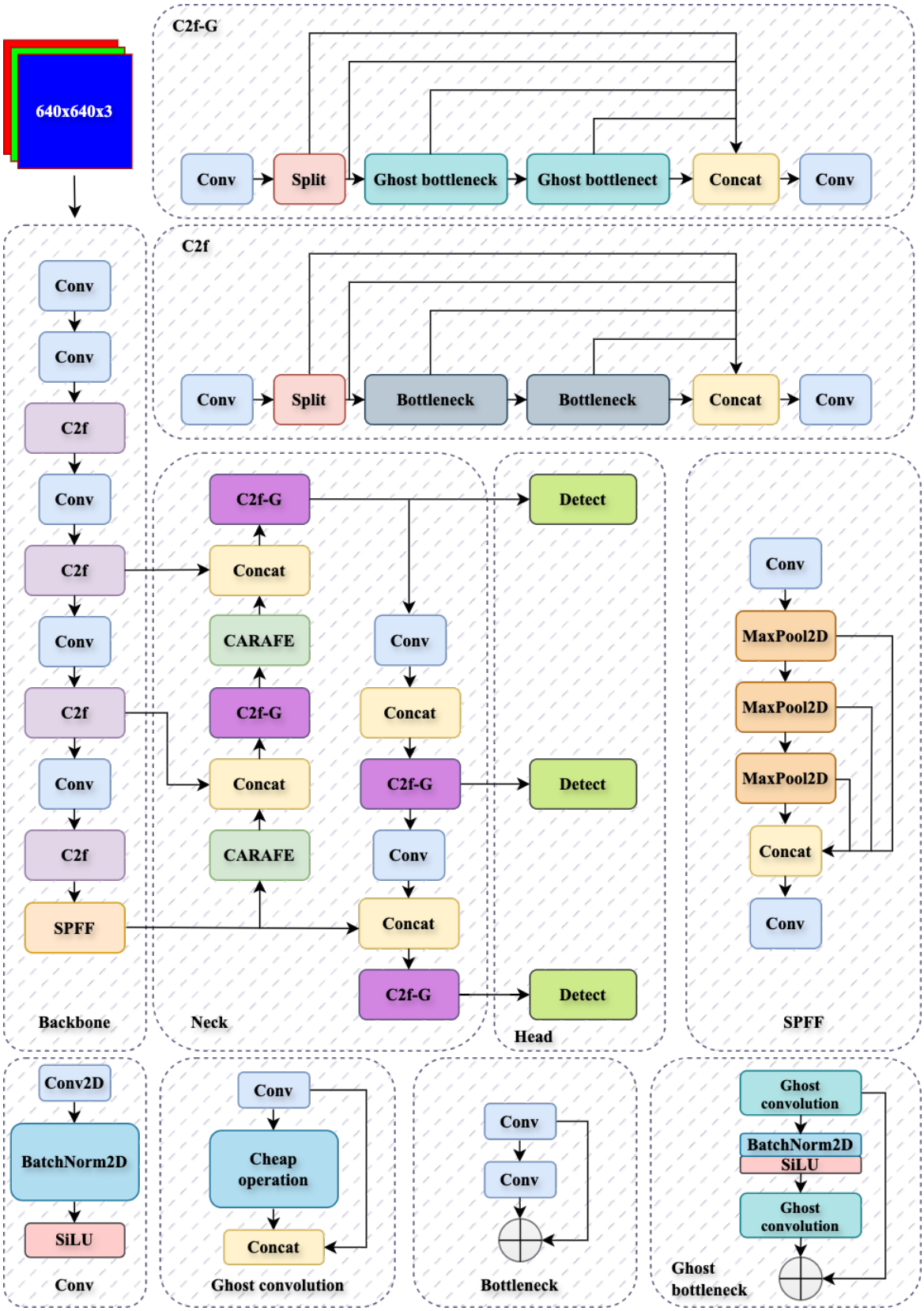


**Figure 2:** Overall schematics of proposed network, including the proposed C2f-G block for better feature representation and CARAFE upsampling operator for better small and overlapping target information processing.

First, we address the high computational complexity of the C2f block due to the excessive number of 3x3 convolutions and in order to mitigate its influence on overall network complexity, we propose a custom block named C2f-G, which serves as a drop-in replacement for the original C2f block. The C2f-G block significantly reduces the number of IO operations by replacing the bottleneck components with GhostBottleneck, derived from GhostNet [32]. GhostBottleneck achieves computational efficiency by generating more feature maps from fewer intrinsic feature maps through cheap linear operations, thus maintaining performance while reducing the parameter count. This modification is crucial for enabling the deployment of the model on edge computing devices without sacrificing accuracy and making room for other improvements, which introduce additional parameters to the object detection system.

Second, we enhance the feature upsampling process in the neck of YOLOv8 by incorporating the Content-Aware ReAssembly of FEatures (CARAFE) operator [33]. Traditional upsampling methods such as nearest neighbor and bilinear interpolation are computationally efficient but result in semantic information loss, which is detrimental in underwater environments where small and occluded objects are prevalent. CARAFE addresses this by using adaptive kernels generated from the input features to perform content-aware reassembly. This process preserves fine-grain details, ensuring that critical information is retained during upsampling. CARAFE's kernel prediction and reassembly modules allow for more accurate and semantically rich feature maps, improving the detection of small and occluded targets.

Finally, to improve the model's performance in detecting overlapping and densely located objects, we replace the traditional Non-Maximum Suppression (NMS) and Distance-IoU NMS (DIoU-NMS) with Soft-NMS [34]. Traditional NMS uses a hard threshold to suppress overlapping bounding boxes, which can lead to the loss of true positive detections. DIoU-NMS, while considering the distance between box centers, still employs a hard threshold mechanism. In contrast, Soft-NMS reduces the confidence scores of overlapping boxes using a Gaussian penalty function, resulting in a more gradual suppression process, improving detection accuracy in scenes with high object density and overlap.

### 4.1. Network lightweighting with C2f-G block

The C2f block contributes a lot to overall network complexity in YOLOv8 due to an excessive number of 3x3 convolutions. This results in a high parameter count, which poses challenges for deploying the model on edge computing devices. To address this, light-weighting the structure of the C2f block is essential. We propose a custom block named C2f-G, which serves as a drop-in replacement for the C2f block in the original YOLOv8 network. C2f-G significantly reduces the number of IO operations by replacing the bottleneck components with GhostBottleneck.

The GhostBottleneck, derived from GhostNet, is an efficient convolutional neural network design that aims to reduce computational complexity while maintaining performance. GhostNet achieves this by generating more feature maps from fewer intrinsic feature maps through series of linear transformations called “cheap operations”, mimicking the behavior of standard convolutions with reduced computational cost.

In a traditional convolution operation, the output feature map $Y$ is generated from the input feature map $X$ using a convolution kernel $W$ as $Y = X * W$ where $*$ denotes the convolution operation. When dealing with high-dimensional data and large kernel sizes, as seen in the 3x3 convolutions prevalent in the C2f block, this quickly becomes computationally expensive. The GhostNet approach to convolution, however, generates the same number of output feature maps with fewer computations. It splits the convolution process into two parts: the primary convolution and the ghost module. The primary convolution uses fewer filters to generate intrinsic feature maps $Y_{\text{int}}$ as $Y_{\text{int}} = X * W_{\text{int}}$ where $W_{\text{int}}$ is a smaller convolutional filter. Then, acquired intrinsic feature maps are processed through a series of cheap operations to generate the ghost feature maps $Y_{\text{ghost}}$ as in $Y_{\text{ghost}} = f(Y_{\text{int}})$ where $f$ denotes cheap operation.

Schematically, Ghost convolution can be drawn as follows:

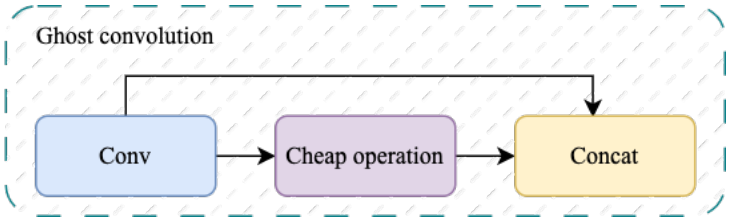


**Figure 3:** Ghost convolution schematics

The Ghost bottleneck module is then composed of two Ghost convolutions with batch normalization and SiLU activation in between.

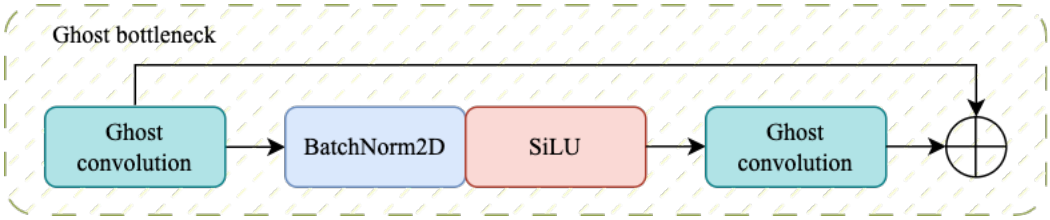


**Figure 4:** Ghost bottleneck structure

C2f-G module includes two convolutional layers with two Ghost bottlenecks in between. Ordinary convolution operation is used, as more representational power is needed to extract features in the first layer of a block.

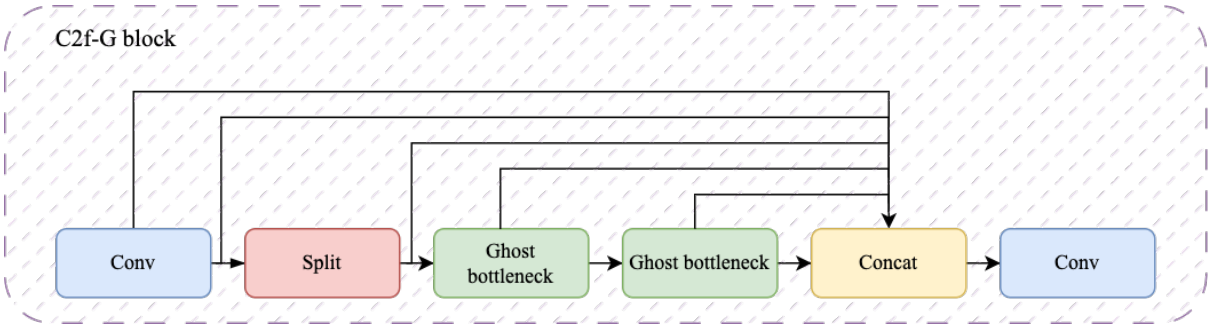


**Figure 5:** C2f-G block elements

C2f-G can achieve the similar representational power as traditional bottlenecks but with significantly fewer parameters and reduced computational complexity, enabling efficient deployment on edge computing devices with minimal effect on model performance.

### 4.2. Enhanced feature upsampling

In YOLO-based detectors, feature maps are obtained from the backbone network and then form a feature pyramid in the neck with upsampling processes in between. The commonly used upsampling methods are nearest neighbor and bilinear interpolation. While these methods are computationally efficient, as they add little to no additional parameters, they result in significant semantic information loss. In the context of underwater images, this loss can lead to the missed detection of small and occluded targets, as critical information about these targets gets diluted during the upsampling process. Therefore, selecting a proper feature upsampling method is crucial to preserve fine-grain information about the targets.

To address this issue, we selected Content-Aware ReAssembly of FEatures (CARAFE) as an effective and performance-friendly feature upsampling operator to replace the standard upsampling

layers in the YOLOv8 neck. CARAFE offers several advantages over traditional methods by leveraging content-aware mechanisms to preserve and enhance semantic information during the upsampling process.

CARAFE works through a two-step process involving a kernel prediction module and a content-aware reassembly module. Given an input feature map $C$ of size $C \times H \times W$ and an upsample ratio $\sigma$, CARAFE produce an output feature map $X'$sized $C \times (\sigma H) \times (\sigma W)$. For any target location $l' = (i', j')$ in the output $X'$ there is a corresponding source location $l = (i, j)$ in the input $X$, where $i = \lfloor i'/\sigma \rfloor$ and $j = \lfloor j'/\sigma \rfloor$. The neighbor of $X_l$ is denoted as $N(X_l, k)$ , which represents the $k \times k$ sub-region centered at $l$.

In the first step, the kernel prediction module $\psi$ predicts a reassembly kernel $W_{l'}$ for each target location $l'$ based on the neighbor of $X_l$:

$$W_{l'} = \psi(N(X_l, k_{\text{encoder}})).$$

In the second step, the content-aware reassembly module $\phi$ reassembles the features using the predicted kernels:

$$X'_{l'} = \phi(N(X_l, k_{\text{up}}), W_{l'}).$$

The kernel prediction module is composed of three sub-modules: the channel compressor, the content encoder, and the kernel normalizer. The channel compressor reduces the channel of the input feature map, making the process more computationally efficient. The content encoder generates the reassembly kernels based on the input features, while the kernel normalizer applies a softmax function to ensure that the reassembly kernels sum to one.

The content-aware reassembly module performs the reassembly operation using a weighted sum

$$X'_{l'} = \sum_{n=-r}^{r} \sum_{m=-r}^{r} W_{l'}(n, m) \cdot X_{(i+n, j+m)},$$

where $r = \lfloor k_{\text{up}}/2 \rfloor$. This approach allows each pixel within the local region to contribute to the upsampled pixel $l'$ based on the content, rather than just the spatial distance.

Improving feature upsampling process in object detection neck with CARAFE operator ensures a more accurate and semantically rich upsampling of feature maps, which is particularly beneficial for detecting small and occluded objects in underwater images.

### 4.3. Overlapping targets processing with Soft-NMS

To improve the performance of our model in detecting overlapping and densely located objects, we chose to employ Soft Non-Maximum Suppression (Soft-NMS) instead of traditional NMS or Distance-IoU NMS (DIoU-NMS). The motivation behind it lies in the specific challenges posed by underwater images, where objects are often small, occluded, and densely packed. Traditional NMS, while effective in reducing redundant detections by suppressing overlapping bounding boxes with lower scores, often leads to the suppression of true positive detections, especially in scenarios with significant object overlap. This occurs because traditional NMS uses a hard threshold to discard boxes, which can result in missing true objects that are close to each other. Similarly, DIoU-NMS, which incorporates the distance between the centers of bounding boxes to improve suppression accuracy, still uses a hard threshold mechanism that can fail in densely populated scenes.

Soft-NMS addresses these issues by modifying the suppression process. Instead of outright discarding overlapping bounding boxes, Soft-NMS decreases their confidence scores based on the degree of overlap. This approach reduces the likelihood of missing true positive detections that are close to each other. Specifically, the pruning step in Soft-NMS involves applying a Gaussian penalty

function to the scores of the overlapping boxes, ensuring a more gradual and continuous suppression. This is particularly beneficial for underwater images, where the presence of dense and overlapping objects is common, and the preservation of potential true positives is crucial for accurate detection.

The mathematical formulation of Soft-NMS can be described as follows. For each detection box $b_i$ with a score $s_i$ and an overlap with the maximum score box $M$, the updated score $s_i'$ is computed using a Gaussian function:

$$s_i' = s_i \cdot e^{-\frac{\text{IoU}(M,b_i)^2}{\sigma}}$$

Here, $\text{IoU}(M, b_i)$ represents the Intersection over Union between the maximum score box $M$ and the detection box $b_i$, and $\sigma$ is parameter that controls the decay rate of the scores. This continuous penalty function ensures that boxes with higher overlap with $M$ receive a greater penalty, thereby reducing their scores more significantly than those with lower overlap. This method allows for a more nuanced suppression strategy, effectively balancing between eliminating false positives and retaining true positives, which is especially critical in environments with high object density and overlap.

Figure 6 demonstrates the results of using Soft-NMS for underwater object detection. As seen in the picture, DIoU-NMS accounts for the distance between the centers of found objects and tends to remove predictions which are too close to each other, which in case with underwater object detection leads to a higher number of false negatives.

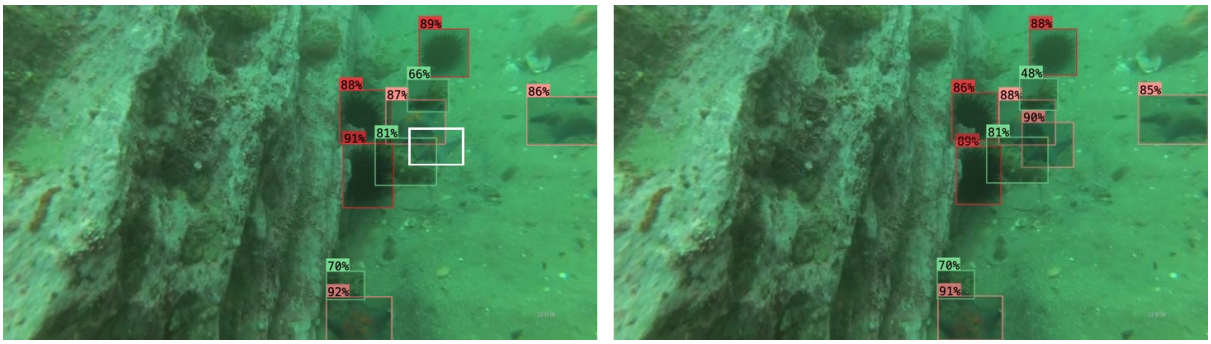


**Figure 6:** Example of DIoU-NMS and Soft-NMS algorithm results. Detections on the left are obtained from a model using DIoU non-maximum suppression strategy, detections of the right are from a model with Soft non-maximum suppression. White colored boxes denote false negatives

## 5. Experiment results

A challenging underwater detection dataset UTDAC2020, which is short for Underwater Target Detection Algorithm Competition 2020, has been selected to test the performance of the proposed algorithm. The dataset features 5168 training and 1293 validation images in various resolutions (3840 x 2160, 1920 x 1080, 720 x 405 and 586 x 480), featuring 4 classes (echinus, holothurian, scallop and starfish). Notably, the dataset has significant class imbalance, with echinus class having four times more samples than starfish, scallop and holothurian. Also, the targets in UTDAC2020 often appear in different scales and are very densely packed, which allows to use this dataset to assess model performance with tiny overlapping targets.

The experimental setup consisted of Intel Core i5-13600K, NVIDIA A4000 GPU with 16GB VRAM. The setup runs Ubuntu 20.04.6 LTS with Python version 3.10.13, CUDA version 12.1, PyTorch version 2.2.1.

The training process ran for 250 epochs with batch size 32 and image size 640 x 640. Stochastic gradient descent (SGD) has been used as an optimization algorithm with momentum 0.937, initial learning rate was 0.01 and weight decay coefficient of 0.005. The optimal set of hyperparameters was

found empirically using Ray Tune library. Default augmentation strategies from Ultralytics YOLOv8 framework have been applied, and no other augmentations have been used.

The metrics to assess model performance are listed in the table below.

**Table 1**
**Metrics used in the experiment**

| Metrics | Description |
|---|---|
| mAp50 | Mean average precision (mAp) at intersection over union (IoU) of 0.50 |
| mAp | mAp at IoU of 0.50:0.05:0.95 |
| Params | Total parameter count of the model |
| FLOPs | Performance metrics denoting number of floating point operations per second |
| Size | Model size in megabytes |

Here, mAp is defined as:

$$mAP = \frac{\sum_{i=1}^{k} AP_i}{K},$$

where $AP$ denotes average precision measured for a current object category, and $K$ denotes the category.

In these experiments, mAp and mAp50 represent the accuracy of neural network, parameter counts and FLOPs represent its complexity, which is crucial for deploying a model on edge hardware, such as integrated compute units of autonomous underwater vehicles. As seen in comparison results, the performance of the proposed approach surpasses even larger YOLOv8l model, which uses 6 times more parameters, making it applicable in real-world applications on edge hardware of autonomous underwater vehicles.

**Table 2**
**Experiment results on UTDAC2020 dataset**

| Method | Backbone | mAp | mAp50 | Params (M) | FLOPs (G) | Size (Mb) |
|---|---|---|---|---|---|---|
| Faster R-CNN | ResNet50 | 44.50 | 80.90 | 41.14 | 63.26 | ~ |
| RetinaNet | ResNet50 | 43.90 | 80.40 | 36.17 | 52.62 | ~ |
| FCOS | ResNet50 | 43.90 | 81.10 | 31.84 | 50.36 | ~ |
| YOLOv8n | DarkNet-53 | 49.01 | 82.65 | 3.0 | 8.1 | 6 |
| YOLOv8s | DarkNet-53 | 50.53 | 84.71 | 11.1 | 28.4 | 22 |
| YOLOv8m | DarkNet-53 | 51.69 | 84.92 | 25.8 | 78.7 | 51 |
| YOLOv8l | DarkNet-53 | 51.71 | 84.97 | 43.6 | 164.8 | 84 |
| YOLOv8s w/FasterNet [28] | FasterNet-T0 | 52.12 | 85.49 | 8.5 | 25.5 | 17 |
| Ours | DarkNet-53 | 52.45 | 86.18 | 9 | 27 | 18 |

To further validate the performance of the proposed approach, additional testing has been conducted on Underwater Robot Picking Contest 2019 (URPC2019) dataset, which includes 3765 training samples and 942 validation samples in the same four categories: echinus, holothurian, scallop and starfish. The class balance is skewed towards echinus class, with other classes featuring three times less samples each. The exact same experiment configuration, hyperparameters and metrics were used to compare our model performance with existing YOLOv8 models. The obtained results prove generalization capabilities of our approach, demonstrating superior underwater object detection capabilities.\

**Table 3**
**Experiment results on URPC2019 dataset**

| Method | Backbone | mAp | mAp50 | Params (M) | FLOPs (G) | Size (Mb) |
|---|---|---|---|---|---|---|
| YOLOv8n | DarkNet-53 | 47.32 | 80.3 | 3.0 | 8.1 | 6 |
| YOLOv8s | DarkNet-53 | 48.81 | 81.42 | 11.1 | 28.4 | 22 |
| YOLOv8m | DarkNet-53 | 48.74 | 81.23 | 25.8 | 78.7 | 51 |
| YOLOv8l | DarkNet-53 | 49.65 | 82.55 | 43.6 | 164.8 | 84 |
| Ours | DarkNet-53 | 51.78 | 85.1 | 9 | 27 | 18 |

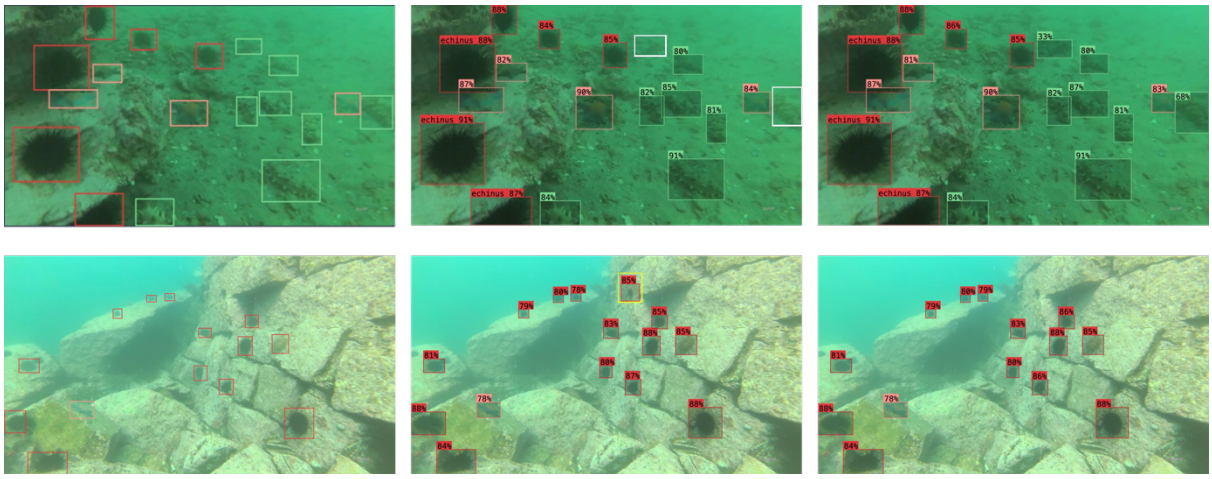

**Figure 7:** From left to right: round truth labels, detections from Ultralytics YOLOv8s model, detections made using our model. White boxes denote false negatives, yellow boxes denote false positives.

## 6. Conclusions

A methodology for resource-efficient detection of small and overlapping targets in underwater images is proposed. Hybrid object detector based on YOLOv8 model is used as a base. The proposed modifications include network light-weighting by introducing C2f-G block, which replaces the original C2f block bottleneck with a more efficient Ghost bottleneck structure. Content-Aware ReAssembly of Features usage is proposed to enhance feature upsampling process in the neck of object detector to prevent fine information loss in feature maps and enhance small target detection. Soft-NMS is employed as non-maximum suppression strategy to handle overlapping and densely located objects bounding box information.

The object detection model based on proposed methodology achieves a mean Average Precision (mAP) of 52.45%, and a mAP at 50% IoU (mAP50) of 86.18%. Notably, the model features only 9 million parameters, 26 billion floating point operations per second (FLOPS), and a size of 18 MB, not only providing better performance than the best existing solutions but also achieving higher accuracy while being six times smaller than YOLOv8l, the larger model from YOLO family.

The proposed applications of the model include deploying it directly on autonomous underwater vehicle hardware for real-time object detection tasks.

## 7. References


[1] M. Zgurovsky, V. Sineglazov, and E. Chumachenko, “Classification and Analysis of Multicriteria Optimization Methods,” in Artificial Intelligence Systems Based on Hybrid Neural Networks, vol. 904, Springer, Cham, 2021, pp. 2. doi: https://doi.org/10.1007/978-3-030-48453-8_2

[2] M. Zgurovsky, V. Sineglazov, and E. Chumachenko, “Classification and Analysis Topologies Known Artificial Neurons and Neural Networks,” in Artificial Intelligence Systems Based on

Hybrid Neural Networks, vol. 904, Springer, Cham, 2021, pp. 1. doi: https://doi.org/10.1007/978-3-030-48453-8_1
[3] M. Zgurovsky, V. Sineglazov, and E. Chumachenko, "Classification and Analysis of Multicriteria Optimization Methods," in Artificial Intelligence Systems Based on Hybrid Neural Networks, vol. 904, Springer, Cham, 2021, pp. 2. doi: https://doi.org/10.1007/978-3-030-48453-8_2
[4] V. Sineglazov, K. Riazanovskiy, A. Klanovets, E. Chumachenko, and N. Linnik, "Intelligent tuberculosis activity assessment system based on an ensemble of neural networks," Comput. Biol. Med., vol. 147, Aug. 2022, Art. no. 105800. doi: https://doi.org/10.1016/j.compbiomed.2022.105800
[5] V. M. Sineglazov, "Multi-functional integrated complex of detection and identification of UAVs," in 2015 IEEE International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD), Kyiv, Ukraine, 2015, pp. 320-323. doi: https://doi.org/10.1109/APUAVD.2015.7346631
[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: https://doi.org/10.1109/tpami.2016.2577031.
[7] W. Liu et al., "SSD: Single Shot MultiBox Detector," Computer Vision – ECCV 2016, vol. 9905, pp. 21–37, 2016, doi: https://doi.org/10.1007/978-3-319-46448-0_2.
[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–1, 2018, doi: https://doi.org/10.1109/tpami.2018.2858826.
[9] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," Jul. 2018, doi: https://doi.org/10.48550/arXiv.2107.08430.
[10] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully Convolutional One-Stage Object Detection," Sep. 2019, doi: https://doi.org/10.48550/arXiv.1904.01355.
[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," arXiv.org, Jun. 2015, doi: https://doi.org/10.48550/arXiv.1506.02640.
[12] Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE, July 2017.
[13] Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv 2018, arXiv:1804.02767.
[14] Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal speed and accuracy of object detection. arXiv 2020, arXiv:2004.10934.
[15] "Ultralytics YOLOv5 - Ultralytics YOLOv8 Docs," docs.ultralytics.com. https://docs.ultralytics.com/yolov5/
[16] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv:2207.02696 [cs], Jul. 2022, Available: https://arxiv.org/abs/2207.02696
[17] J. Solawetz, "What is YOLOv8? The Ultimate Guide.," Roboflow Blog, Jan. 11, 2023. https://blog.roboflow.com/whats-new-in-yolov8/
[18] M. Y. Ouis and M. Akhloufi, "YOLO-Based Fish Detection in Underwater Environments", in ECRS 2023. Basel Switzerland: MDPI, 2023. Accessed: Jun. 3, 2024. [Online]. Available: https://doi.org/10.3390/ecrs2023-16315
[19] S. Fang, L. Mu, S. Jia, K. Liu, and D. Liu, "Research on sunken & submerged oil detection and its behavior process under the action of breaking waves based on YOLO v4 algorithm", Mar. Pollut. Bull., vol. 179, p. 113682, Jun. 2022. Accessed: Jun. 1, 2024. [Online]. Available: https://doi.org/10.1016/j.marpolbul.2022.113682
[20] C. Wu, Y. Sun, T. Wang, and Y. Liu, "Underwater trash detection algorithm based on improved YOLOv5s", J. Real-Time Image Process., Jul. 2022. Accessed: Jun. 2, 2024. [Online]. Available: https://doi.org/10.1007/s11554-022-01232-0
[21] U. N. Dulhare and M. Hussam Ali, "Underwater human detection using faster R-CNN with data augmentation", Mater. Today: Proc., Jun. 2021. Accessed: Jun. 2, 2024. [Online]. Available: https://doi.org/10.1016/j.matpr.2021.05.653

[22] M. Zhang, S. Xu, W. Song, Q. He, and Q. Wei, “Lightweight Underwater Object Detection Based on YOLO v4 and Multi-Scale Attentional Feature Fusion,” Remote Sensing, vol. 13, no. 22, p. 4706, Nov. 2021, doi: https://doi.org/10.3390/rs13224706.
[23] K. Liu, L. Peng, and S. Tang, “Underwater Object Detection Using TC-YOLO with Attention Mechanisms,” Sensors, vol. 23, no. 5, p. 2567, Jan. 2023, doi: https://doi.org/10.3390/s23052567.
[24] X. Shen, X. Sun, H. Wang, and X. Fu, “Multi-dimensional, multi-functional and multi-level attention in YOLO for underwater object detection,” Neural Computing and Applications, vol. 35, no. 27, pp. 19935–19960, Jul. 2023, doi: https://doi.org/10.1007/s00521-023-08781-w.
[25] F. Xu, H. Wang, J. Peng, and X. Fu, “Scale-aware feature pyramid architecture for marine object detection,” Neural Computing and Applications, vol. 33, no. 8, pp. 3637– 3653, Jul. 2020, doi: https://doi.org/10.1007/s00521-020-05217-7.
[26] T.-S. Pan, H.-C. Huang, J.-C. Lee, and C.-H. Chen, “Multi-scale ResNet for real- time underwater object detection,” Signal, Image and Video Processing, vol. 15, no. 5, pp. 941–949, Nov. 2020, doi: https://doi.org/10.1007/s11760-020-01818-w.
[27] Z. Wang, G. Zhang, K. Luan, C. Yi, and M. Li, “Image-Fused-Guided Underwater Object Detection Model Based on Improved YOLOv7,” Electronics, vol. 12, no. 19, pp. 4064–4064, Sep. 2023, doi: https://doi.org/10.3390/electronics12194064.
[28] M. Zhang, Z. Wang, W. Song, D. Zhao, and H. Zhao, “Efficient Small-Object Detection in Underwater Images Using the Enhanced YOLOv8 Network,” Applied Sciences, vol. 14, no. 3, p. 1095, Jan. 2024, doi: https://doi.org/10.3390/app14031095.
[29] A. Guo, K. Sun, and Z. Zhang, “A lightweight YOLOv8 integrating FasterNet for real-time underwater object detection,” Journal of real-time image processing, vol. 21, no. 2, Mar. 2024, doi: https://doi.org/10.1007/s11554-024-01431-x.
[30] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 07, pp. 12993–13000, Apr. 2020, doi: https://doi.org/10.1609/aaai.v34i07.6999.
[31] Sineglazov V., Savchenko M. A Comprehensive Framework for Underwater Object Detection Based on Improved YOLOv8. Electronics and Control Systems. 2024. Vol. 1, no. 79. P. 9–15. URL: https://doi.org/10.18372/1990-5548.79.18429(date of access: 23.06.2024).
[32] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, “GhostNet: More Features From Cheap Operations”, in 2020 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, USA, Jun. 13–19, 2020. IEEE, 2020. Accessed: Jun. 2, 2024. [Online]. Available: https://doi.org/10.1109/cvpr42600.2020.00165
[33] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, “CARAFE++: Unified Content-Aware ReAssembly of FEatures”, IEEE Trans. Pattern Anal. Mach. Intell., p. 1, 2021. Accessed: Jun. 5, 2024. [Online]. Available: https://doi.org/10.1109/tpami.2021.3074370
[34] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, “Soft-NMS — Improving Object Detection with One Line of Code”, in 2017 IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Oct. 22–29, 2017. IEEE, 2017. Accessed: Aug. 6, 2024. [Online]. Available: https://doi.org/10.1109/iccv.2017.593